CLAIMS

1. One or more processor-accessible media comprising processor-executable instructions that, when executed, direct a device to perform actions comprising:

accepting a connection;

aggregating a connection state for the connection from a protocol stack; and

sending the connection state.

2. The one or more processor-accessible media as recited in claim 1, wherein the action of accepting comprises an action of:

sending an acknowledgment packet in response to a connection-requesting packet.

3. The one or more processor-accessible media as recited in claim 1, comprising the processor-executable instructions that, when executed, direct the device to perform a further action comprising:

receiving data for the connection;

wherein the action of aggregating comprises an action of:

aggregating the connection state from a protocol state of the protocol stack and the data.

4. The one or more processor-accessible media as recited in claim 1, wherein the action of aggregating comprises an action of:

compiling a protocol state from the protocol stack.

5. The one or more processor-accessible media as recited in claim 4, wherein the action of compiling comprises an action of:

compiling the protocol state from the protocol stack starting at a highest level of the protocol stack.

6. The one or more processor-accessible media as recited in claim 4, wherein the action of compiling comprises an action of:

compiling the protocol state from the protocol stack at a transmission control protocol (TCP) stack portion and an internet protocol (IP) stack portion.

7. The one or more processor-accessible media as recited in claim 1, wherein the action of sending comprises actions of:

bundling the connection state with a flow identifier that corresponds to the connection to produce a binary blob; and

transmitting the binary blob from an originating device to a target device.

8. The one or more processor-accessible media as recited in claim 1, wherein the action of sending comprises actions of:

bundling the connection state with a flow identifier that corresponds to the connection to produce a binary blob; and

transmitting the binary blob from an originating device to a target device in a reliable manner such that the binary blob may be received intact at the target device even if one or more packets that comprise the binary blob are lost or corrupted.

9. The one or more processor-accessible media as recited in claim 1, comprising the processor-executable instructions that, when executed, direct the device to perform further actions comprising:

selecting a flow identifier for the connection responsive to a connection counter; and

sending the flow identifier to identify packets corresponding to the connection.

10. The one or more processor-accessible media as recited in claim 1, wherein the action of sending comprises an action of:

sending the connection state to a targeted device;

wherein the processor-executable instructions, when executed, direct the device to perform a further action comprising:

forwarding subsequent packets for the connection to the targeted device using a flow identifier to encapsulate the subsequent packets.

11. One or more processor-accessible media comprising processor-executable instructions that, when executed, direct a device to perform actions comprising:

receiving a connection state for a connection;

injecting the connection state for the connection into a network stack; and

continuing the connection using the injected connection state.

12. The one or more processor-accessible media as recited in claim 11, wherein the action of continuing comprises an action of:

continuing the connection by indicating received packets up to an application in accordance with the injected connection state.

13. The one or more processor-accessible media as recited in claim 11, wherein:

the action of receiving comprises an action of:

receiving the connection state, the connection state having a protocol state and data for the connection; and

the action of injecting comprises an action of:

injecting the protocol state into a protocol stack portion of the network stack.

14. The one or more processor-accessible media as recited in claim 13, wherein the action of injecting the connection state further comprises an action of:

indicating the data for the connection up the network stack toward an application.

15. The one or more processor-accessible media as recited in claim 11, wherein the action of injecting comprises an action of:

infusing a protocol state from the connection state into a protocol stack portion of the network stack.

16. The one or more processor-accessible media as recited in claim 15, wherein the action of infusing comprises an action of:

infusing the protocol state into the protocol stack starting at a highest level of the protocol stack.

17. The one or more processor-accessible media as recited in claim 15, wherein the action of infusing comprises an action of:

infusing the protocol state into the protocol stack at a transmission control protocol (TCP) stack portion and an internet protocol (IP) stack portion.

18. The one or more processor-accessible media as recited in claim 11, wherein the action of receiving comprises actions of:

receiving a binary blob from an originating device at a target device, the binary blob including the connection state and a flow identifier that corresponds to the connection; and

unbundling the connection state and the flow identifier at a level of the network stack that is below a protocol stack portion of the network stack.

19. The one or more processor-accessible media as recited in claim 11, comprising the processor-executable instructions that, when executed, direct the device to perform further actions comprising:

receiving an encapsulation mapping; and

storing the received encapsulation mapping in an encapsulation mapping table that may be accessed according to flow identifier.

20.   The one or more processor-accessible media as recited in claim 11, wherein the action of receiving comprises an action of:

receiving the connection state from an originating device;

wherein the processor-executable instructions, when executed, direct the device to perform a further action comprising:

receiving from the originating device encapsulated packets that have a flow identifier; and

de-encapsulating the encapsulated packets using an encapsulation mapping entry that links the flow identifier to a source/destination pair.

21.   One or more processor-accessible media comprising processor-executable instructions that comprise at least part of an application, wherein the processor-executable instructions, when executed, enable the application to initiate a connection migration by issuing a migrate connection function call to a socket layer such that the socket layer propagates the migrate connection function call to a protocol stack, the migrate connection function call precipitating at the protocol stack a compilation of protocol state from the protocol stack.

22.   The one or more processor-accessible media as recited in claim 21, wherein the application comprises at least one of a general application, a hosted application, and a load balancing application.

23. A device comprising:

a connection migrator that is configured to migrate connections away from the device; the connection migrator capable of precipitating a compilation of protocol state for a connection across a protocol stack; the connection migrator adapted to aggregate the compiled protocol state with data for the connection into an aggregated connection state of the connection; the connection migrator further capable of causing the aggregated connection state to be sent toward a target device.

24. The device as recited in claim 23, wherein the connection migrator is realized at least partially in software.

25. The device as recited in claim 23, wherein the device further comprises:

a classifier that is capable of accepting the connection, the classifier adapted to issue a migrate connection command to the connection migrator.

26. The device as recited in claim 25, wherein the data, which is aggregated with the compiled protocol state into the aggregated connection state, comprises data that has been acknowledged by the classifier.

138

27. The device as recited in claim 23, wherein the connection migrator comprises:

a migrator shim that is located above the protocol stack in a network stack of the device; and

a migrator intermediate driver that is located below the protocol stack in the network stack of the device.

28. The device as recited in claim 27, wherein:

the migrator shim is located between the protocol stack and a socket layer of the network stack of the device; and

the migrator intermediate driver is located between the protocol stack and at least one miniport of the network stack of the device.

29. The device as recited in claim 28, wherein the migrator intermediate driver is located at a protocol-hardware interface layer of the network stack of the device.

30. The device as recited in claim 27, wherein:

the migrator shim is adapted to receive a migrate connection command and to propagate the migrate connection command to the protocol stack; and

the migrator intermediate driver is adapted to divert a copy of the data for the connection for subsequent aggregation with the compiled protocol state.

31. The device as recited in claim 27, wherein the migrator intermediate driver is adapted to bundle the aggregated connection state and a flow identifier for the connection into a binary blob.

32. The device as recited in claim 23, wherein the connection comprises a transmission control protocol/internet protocol (TCP/IP) connection, and the compiled protocol state includes information related to a TCP/IP connection.

33. The device as recited in claim 23, wherein the device further comprises:

the protocol stack, the protocol stack adapted to compile the protocol state for the connection into the compiled protocol state responsive to a migrate connection command.

34. A device comprising:

a connection migrator that is configured to migrate connections onto the device; the connection migrator adapted to intercept a connection state for a connection that is sent from an originating device, the connection state including a protocol state and data; the connection migrator further adapted to inject the connection state into a network stack of the device; the connection migrator capable of precipitating an infusion of the protocol state for the connection across a protocol stack of the network stack.

35.   The device as recited in claim 34, wherein the connection migrator comprises at least part of load balancing infrastructure that is resident at and executing on the device.

36.   The device as recited in claim 34, wherein the device further comprises:

an application that is hosted on the device;

wherein the connection migrator is further configured to migrate connections onto the device in a manner that is transparent to the application.

37.   The device as recited in claim 34, wherein the data included in the connection state comprises data that was acknowledged at the originating device.

38.   The device as recited in claim 34, wherein the connection migrator comprises:

a migrator shim that is located above the protocol stack in the network stack of the device; and

a migrator intermediate driver that is located below the protocol stack in the network stack of the device.

141

39. The device as recited in claim 38, wherein:

the migrator shim is located between the protocol stack and a socket layer of the network stack of the device; and

the migrator intermediate driver is located between the protocol stack and at least one miniport of the network stack of the device.

40. The device as recited in claim 38, wherein:

the migrator shim is adapted to be notified by the migrator intermediate driver of a migration uploading procedure and is adapted to initiate a protocol state infusion routine with the protocol stack; and

the migrator intermediate driver is adapted to detect arrival of the connection state for the connection, to divert the connection state away from a lower portion of the protocol stack, and to notify the migrator shim of the migration uploading procedure.

41. The device as recited in claim 38, wherein the migrator intermediate driver is adapted to un-bundle a received binary blob that includes the connection state for the connection and a flow identifier corresponding to the connection.

42. The device as recited in claim 34, wherein the device further comprises:

the protocol stack, the protocol stack including a transmission control protocol (TCP) layer and an internet protocol (IP) layer.

43. The device as recited in claim 34, wherein the device further comprises:

the protocol stack, the protocol stack adapted to infuse the protocol state for the connection across the protocol stack responsive to initiation of an infuse protocol state routine.

44. An arrangement for connection manipulation, comprising:

migration means for migrating connections from an originating device to a target device by transferring a connection state for a connection being migrated from the originating device to the target device; and

tunneling means for tunneling packets for migrated connections in an encapsulated format from the originating device to the target device.

45. The arrangement as recited in claim 44, wherein the migration means comprises:

aggregation means for aggregating the connection state from a protocol state and from data acknowledged for the connection being migrated.

46. The arrangement as recited in claim 45, wherein the aggregation means comprises:

compilation means for compiling the protocol state from a protocol stack.

47.   The arrangement as recited in claim 44, wherein the tunneling means comprises:

encapsulation means for encapsulating the packets for the migrated connections using flow identifiers that replace at least a part of source/destination address information pairs of the packets.

48.   The arrangement as recited in claim 44, wherein the tunneling means comprises:

mapping means for mapping individual source/destination address information pairs to individual flow identifiers, respective source/destination address information pairs identifying respective migrated connections.

49.   The arrangement as recited in claim 44, wherein the migration means comprises:

driver means for bundling the connection state with a flow identifier that is usable for encapsulating packets of the connection being migrated.

50.   The arrangement as recited in claim 44, wherein the migration means comprises:

injection means for injecting a protocol state portion of the connection state for the connection being migrated into a protocol stack portion of a network stack.

51.   The arrangement as recited in claim 44, wherein the tunneling means comprises:

de-encapsulation means for de-encapsulating the tunneled packets for the migrated connections using flow identifiers that link to at least a part of source/destination address information pairs of the packets.

52.   The arrangement as recited in claim 44, wherein the tunneling means comprises:

mapping means for mapping individual flow identifiers to individual source/destination address information pairs, respective source/destination address information pairs identifying respective migrated connections.

53.   The arrangement as recited in claim 44, wherein the migration means comprises:

driver means for diverting the tunneled packets for the migrated connections in the encapsulated format and directing them to the tunneling means.

54.   The arrangement as recited in claim 44, wherein the arrangement comprises at least one device.

55.   The arrangement as recited in claim 44, wherein the arrangement comprises one or more processor-accessible media.

56. A device comprising:

a tunneler that is configured to tunnel packets away from the device; the tunneler having access to an encapsulation mapping table, the encapsulation mapping table including a plurality of encapsulation mapping entries, each encapsulation mapping entry linking at least a portion of a source/destination pair to a flow identifier; the tunneler adapted to accept a packet having a particular source/destination pair; the tunneler capable of looking up a particular flow identifier at a particular encapsulation mapping entry using at least a portion of the particular source/destination pair; wherein the tunneler is further adapted to encapsulate the packet by replacing part of the packet with the particular flow identifier.

57. The device as recited in claim 56, wherein the tunneler is realized at least partially in software.

58. The device as recited in claim 56, wherein the device further comprises:

a forwarder that is capable of receiving the packet from a client, the forwarder adapted to provide the packet to the tunneler.

59. The device as recited in claim 56, wherein the tunneler is further adapted to encapsulate the packet by replacing at least part of the particular source/destination pair of the packet with the particular flow identifier.

60. The device as recited in claim 56, wherein the flow identifier comprises 32 bits.

61. The device as recited in claim 56, wherein the particular source/destination pair comprises a particular source/destination address information pair.

62. A device comprising:

a tunneler that is configured to tunnel packets into the device; the tunneler having access to an encapsulation mapping table, the encapsulation mapping table including a plurality of encapsulation mapping entries, each encapsulation mapping entry linking a flow identifier to at least a portion of a source/destination pair; the tunneler adapted to accept an encapsulated packet having a particular flow identifier; the tunneler capable of looking up a particular source/destination pair at a particular encapsulation mapping entry using the particular flow identifier; wherein the tunneler is further adapted to de-encapsulate the encapsulated packet by replacing the particular flow identifier with at least part of the particular source/destination pair.

63. The device as recited in claim 62, wherein the tunneler comprises a migrator intermediate driver that is adapted to intercept the encapsulated packet to prevent the encapsulated packet from being provided to a protocol stack of the device.

64. The device as recited in claim 62, wherein the device further comprises:

a protocol stack;

wherein the tunneler is further adapted to de-encapsulate the encapsulated packet by replacing the particular flow identifier with at least part of the particular source/destination pair to produce a de-encapsulated packet; the tunneler capable of indicating up the de-encapsulated packet to the protocol stack.

65. The device as recited in claim 64, wherein the tunneler comprises a virtual network adapter that is bound to a destination address of the de-encapsulated packet.

66. The device as recited in claim 62, wherein the device comprises a host of a cluster of hosts.

67. The device as recited in claim 62, wherein the particular source/destination pair corresponds to a transmission control protocol/internet protocol (TCP/IP) 4-tuple; and wherein the tunneler is further adapted to de-encapsulate the encapsulated packet by replacing the particular flow identifier with a source TCP port and a destination TCP port from the TCP/IP 4-tuple.

148

68. One or more processor-accessible media comprising processor-executable instructions that, when executed, direct a device to perform actions comprising:

obtaining at least a portion of a source/destination pair from an incoming packet;

accessing an encapsulation mapping table using the obtained at least a portion of the source/destination pair to locate an encapsulation mapping entry;

extracting a flow identifier from the located encapsulation mapping entry; and

replacing part of the incoming packet with the extracted flow identifier to produce an encapsulated packet.

69. The one or more processor-accessible media as recited in claim 68, comprising the processor-executable instructions that, when executed, direct the device to perform a further action comprising:

receiving the incoming packet from a client.

70. The one or more processor-accessible media as recited in claim 68, comprising the processor-executable instructions that, when executed, direct the device to perform a further action comprising:

routing the encapsulated packet to a host.

71. The one or more processor-accessible media as recited in claim 68, wherein the source/destination pair of the incoming packet comprises a source/destination address information pair of the incoming packet.

72. The one or more processor-accessible media as recited in claim 68, wherein at least a portion of the processor-executable instructions comprise at least part of load balancing infrastructure.

73. The one or more processor-accessible media as recited in claim 68, wherein the action of replacing comprises an action of:

replacing at least part of the source/destination pair with the extracted flow identifier to produce the encapsulated packet.

74. The one or more processor-accessible media as recited in claim 68, wherein:

the action of obtaining comprises an action of:

obtaining at least a portion of a transmission control protocol/internet protocol (TCP/IP) 4-tuple from the incoming packet; and

the action of replacing comprises an action of:

replacing a source TCP port and a destination TCP port from the incoming packet with the extracted flow identifier.

75. The one or more processor-accessible media as recited in claim 74, comprising the processor-executable instructions that, when executed, direct the device to perform a further action comprising:

replacing a source IP address and a destination IP address from the incoming packet with an IP address of an originating device and an IP address of a target device, respectively.

76. One or more processor-accessible media comprising processor-executable instructions that, when executed, direct a device to perform actions comprising:

obtaining a flow identifier from an encapsulated packet;

accessing an encapsulation mapping table using the obtained flow identifier to locate an encapsulation mapping entry;

extracting at least a portion of a source/destination pair from the located encapsulation mapping entry, the source/destination pair identifying a flow of packets for a connection; and

replacing the flow identifier of the encapsulated packet with the extracted at least a portion of the source/destination pair to produce a de-encapsulated packet for the connection.

77. The one or more processor-accessible media as recited in claim 76, comprising the processor-executable instructions that, when executed, direct the device to perform a further action comprising:

receiving the encapsulated packet from load balancing infrastructure.

78. The one or more processor-accessible media as recited in claim 76, comprising the processor-executable instructions that, when executed, direct the device to perform a further action comprising:

indicating the de-encapsulated packet up to a protocol stack.

79. The one or more processor-accessible media as recited in claim 78, wherein the action of indicating comprises an action of:

indicating the de-encapsulated packet up on a virtual network adapter that is bound to a destination address of the de-encapsulated packet.

80. The one or more processor-accessible media as recited in claim 76, comprising the processor-executable instructions that, when executed, direct the device to perform further actions comprising:

receiving an encapsulation mapping; and

storing the encapsulation mapping as the encapsulation mapping entry in the encapsulation mapping table.

81. The one or more processor-accessible media as recited in claim 76, wherein at least a portion of the processor-executable instructions comprise at least part of a migrator intermediate driver that is positioned below a protocol stack portion of a network stack.

82.   The one or more processor-accessible media as recited in claim 76, wherein the source/destination pair of the located encapsulation mapping entry comprises a source/destination address information pair of the located encapsulation mapping entry.

83.   The one or more processor-accessible media as recited in claim 76, wherein the flow of packets for the connection comprises a connection that has been migrated.

84.   The one or more processor-accessible media as recited in claim 76, wherein the action of extracting comprises an action of:

>   extracting a source internet protocol (IP) address, a destination IP address, a source transmission control protocol (TCP) port, and a destination TCP port from the located encapsulation mapping entry.

85.   The one or more processor-accessible media as recited in claim 76, wherein the action of replacing comprises an action of:

>   replacing the flow identifier of the encapsulated packet with a source transmission control protocol (TCP) port and a destination TCP port from the located encapsulation mapping entry to produce the de-encapsulated packet for the connection.

153

86. The one or more processor-accessible media as recited in claim 76, wherein the action of replacing comprises an action of:

replacing an originating internet protocol (IP) address and a destination IP address of the encapsulated packet with a source IP address and a destination IP address, respectively, from the located encapsulation mapping entry to produce the de-encapsulated packet for the connection.